Transaction Management

What is Transaction

- It is a unit of Program that is in a Running Condition.
- A set of changes that must all be made together.
- It is a program unit whose execution may or may not change the Contents of Database.
- It is executed as a single unit.

Example

A transfer of money from one bank account to another requires 2 changes to the database i.e. both must Succeed or Fail together.

For e.g. ATM money transfer (UPDATE commands to increase/decrease Balance).

Processes of Transaction

- Transaction is executed as a series of Reads and Writes of database objects which are:
  - Read Operation
  - Write Operation

Processes of Transaction…

- Read Operation

  Data object is first brought into Main Memory from Disk, and then value is copied into a Program.

- Write Operation

  Data base object copy in Main Memory is first modified then written to Disk.

Transaction Properties

- Atomicity A
- Consistency C
- Isolation I
- Durability D

Transaction Properties…

- Atomicity: (all or nothing)
  - A transaction is said to be Atomic if a transaction always executes all its actions in 1 step or not executes any actions at all.
  - It means either all or none of the transactions operations are performed.

Transaction Properties…

- Consistency: (No violation of Integrity constraints)
  - A transaction must preserve the ***Consistency*** of a database after the execution.
  - This property holds for each transaction.

Transaction Properties…

- Isolation: (concurrent changes invisibles)
  - If several transactions are executed concurrently the results must be same as if they were executed serially in some order.
  - The data used during the execution of a transaction cannot be used by a 2nd transaction until the 1st one complete.

Transaction Properties…

- Durability: (committed update persist)
  - It means once a transaction Commits, the system must guarantee that the results of its operations will never be lost, in spite of some other failures.

Example

Let T1 be a transaction that transfers Rs. 50 from A/C A to A/C B.

A/C= 1000 A/C B=2000

T1 Schedule

Read (A, a)

a=a-50

Write (A, a)

Read (B, b)

B=b+50

Write (B, b)

Power failure occur during transaction T1 is executing and T1has not committed.

Power failure occur after completion of Write (A, a) but before Write (B, b)

i.e. Changes in A are performed but not in B.

Now A=Rs.950 B=Rs.2000

Rs. 50 has been lost due to failure.

Atomicity Case

To maintain atomicity of transaction, database system keeps track of old values of any write operation and if the transaction does not complete its execution, the old values are restored to make it appear as the transaction never executed.

Consistency Case

To maintain consistency of a transaction, the A/C A and A/C B should not be changed.

Isolation Case

To maintain this property, data used during the execution of a transaction cannot be used by a second transaction until the first one is completed.

Solution to Isolation case

Execute each transaction serially one after the other.

Durability

To maintain this property, once a transaction completes successfully, all updates carried out on the database persist, even if there is a system failure after the transaction completes the execution.

States of Transaction

- Active
- Partially Committed
- Failed
- Aborted
- Committed

States of Transaction…

- Active-The Initial state i.e. state while executing.
- Partially Committed- After the Final Statement has been executed.
- Failed-When the Normal execution can no longer proceed.
- Aborted-After the transaction has Rolled back and database has been Restored to its state prior to the start of the transaction.
- Committed-After successful completion.

States of Transaction…

- A transaction has Committed only if it has entered the Committed State.
- A transaction has Aborted only if it has entered the Aborted State.
- A transaction is said to have terminated if it has either Committed or Aborted.

Advantages of Concurrent Execution of Transaction

- Improved Throughput

  Average no. of transactions completed in a given time (CPU Waiting).

- Reduced Waiting Time

  Short transaction can complete quickly.

Scheduling of Transactions

- Complete Schedule
- Serial Schedule
- Non-Serial Schedule
- Equivalent Schedule
- Serializable Schedule

Conflicts of Operations

- Those Read/Write operations whose order cannot be changed without affecting the Consistency of data are called Conflict Operations.
- Some rules for conflict operations
  - If 2 transactions only Read a data object, they do not conflict.
  - If 2 transactions either Read or Write completely separate data objects, they do not conflict.
  - If 1 transaction Writes a data object and another either Reads or Writes the same data objects.

The Inconsistent Analysis Problem

- This problem occurs when a transaction reads several values from a database while a second transaction updates some of them.

  For e.g. Values of variable A, B, C and Sum are in column 3,4,5 and 6. The initial values of A, B, C and Sum is 100, 50, 25 and 0 respectively.

T1	T2	A	B	C	SUM	Remarks
R(A,a)	R(A,a)	Rs.100	Rs.50	Rs.25	0	
sum=sum+A	A=A-10	Rs.100	Rs.50	Rs.25	100	**Value of Sum is changed due to T1 operation and content of local variable 'a' is changed to 90**
R(B,b)	W(A,a)	Rs.90	Rs.50	Rs.25	100	**Write operation on A is performed by T2, so A=90**
sum=sum+B	R(C,c)	Rs.90	Rs.50	Rs.25	150	**Value of Sum is changed due to T1 Operation.**
	c=c+10	Rs.90	Rs.50	Rs.25	150	**Content of local variable 'c' is changed to 35**
	W(C,c)	Rs.90	Rs.50	Rs.35	150	**WRITE operation on C is performed by T2 i.e. = 35**
R(C,c)		Rs.90	Rs.50	Rs.35	150	**Value of C is read out as 35 by T1**
sum=sum+C		Rs.90	Rs.50	Rs.35	185	**Value of Sum is changed due to T1 operation i.e. 185**